



## TECHNICAL ARCHITECTURE (TA) REVIEW PROCESS

October 17, 2008

Prepared by: Robert Woolley, Chief Technologist and Strategic Planner  
Dave Fletcher, Chief Technology Officer

### INTRODUCTION

The TA Review process is a review of project architecture during the design and development phases of an application or infrastructure implementation project. Reviews ensure that the proposed system or infrastructure fits into the existing State environment as well as fitting the future architectural vision. TA Reviews are completed by DTS staff with relevant architecture experience across the State. This review process provides assistance in leveraging the existing architecture, a statewide context for project teams, and identifies opportunities for leveraging other State resources and common services. TA reviews are conducted under the direction of the Architecture Review Board (ARB).

TA reviews are conducted based upon requests from:

- Senior management;
- Project Management Office (PMO);
- Solutions Delivery management;
- Service Delivery management;
- IT Directors;
- Infrastructure and engineering teams;
- Agency business management; and
- Changepoint will also provide automatic ARB notification for project candidates that:
  - Require new infrastructure;
  - New application development;
  - Represent pilot and/or prototype projects;
  - Provide functionality that could potentially be reused by other agencies;
  - Have the potential to set standards as current or future enterprise components;
  - That require new DTS components, services, or expertise;

## Approval Draft 1.5

Projects flagged by Changepoint are candidates for architectural reviews but the reviews will primarily be conducted based upon recommendations from IT Directors and the ARB, and only if such a review adds value.

Projects that are still in initial planning stages are primary candidates for review.

The architecture review process, and the associated TA review document deliverable, has the following purposes in relationship to specific IT projects:

- Ensure that the project architecture is consistent with existing architectures to minimize future costs and changes.
- Ensure the application of best practices concerning architecture work on the project.
- Provide an overview of the project architecture and its compliance with existing enterprise standards.
- Identify where existing standards may require modification.
- Identify services that may be application specific within the project that might be provided as part of enterprise infrastructure.
- Document strategies for resource sharing, collaboration, and other synergies across multiple project and architecture teams.
- Identify and take advantage of advancements to technology.
- Communicate to management the technical readiness of a project.
- Provide a discussion and decision making platform for deciding between architecture alternatives in terms of what is best for the business.
- Provide documentation and a measurable deliverable to the CIO for the decision making purposes.
- Provide an engagement process with new and ongoing development and infrastructure projects that might otherwise be missed.
- Demonstrate rapid and positive support to agency business leaders concerning:
  - business alignment of IT projects;
  - relevance of architecture to business, rather than just the traditional technical architecture perspective; and,

- highlight the main risks for system owners as opposed to the more common technical risk analysis.
- Identify non-compliance mechanisms for:
  - areas to be addressed during realignment;
  - areas for possible consolidation and integration with existing architectures; and,
  - documentation for TA exceptions for non-compliant architectures and standards.
- Timing TA reviews with project lifecycles needs to consider:
  - project initiation;
  - initial design;
  - major design changes; and,
  - ad hoc reviews as requested.

TA reviews consider issues of conformance and compliance, as illustrated in Figure 1.

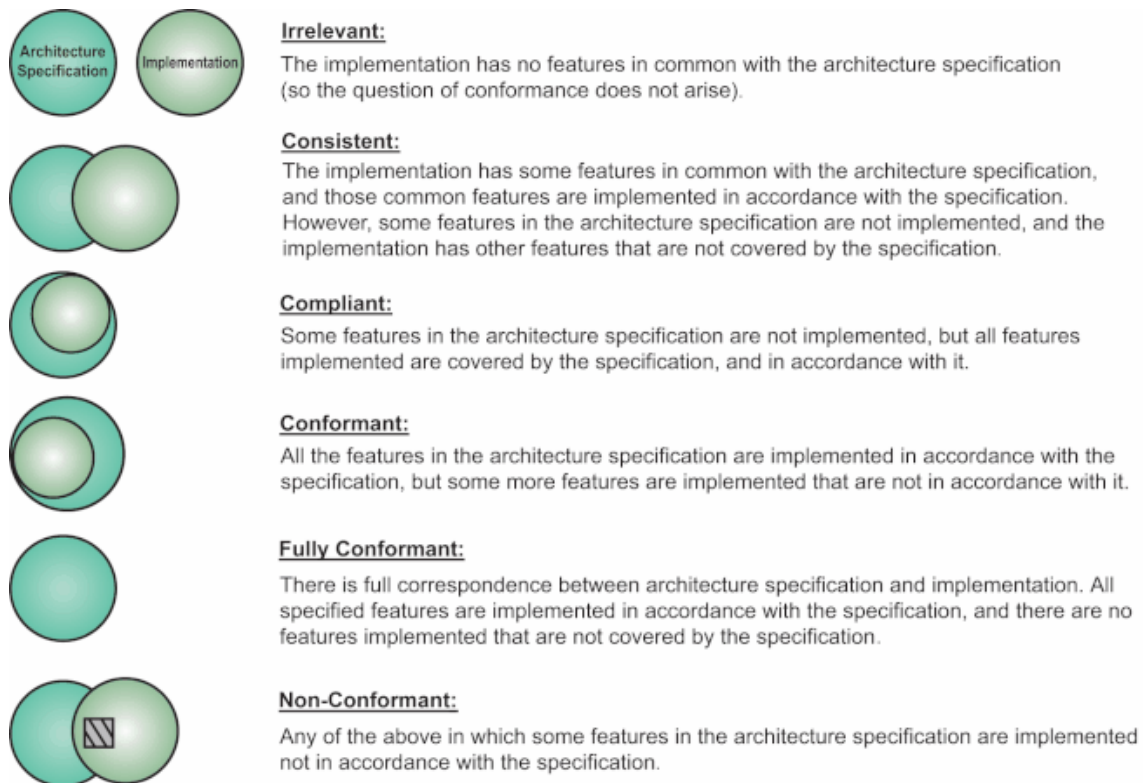


Figure 1. Levels of TA Conformance

The phrase "In accordance with" in Levels of TA Conformance means that the project:

- supports stated strategies and future directions;
- adheres to the stated standards (including syntax and semantic rules specified);
- provides the stated functionality; and,
- adheres to the stated principles (for example: open standards wherever possible and appropriate, and re-use of component building blocks wherever possible and appropriate).

TA conformance is not meant as a heavy handed process, but is focused on delivering value, reduced complexity, and the leveraging of TA resources for the State.

## PROCESS OVERVIEW

Figure 2 illustrates the overall TA review process. The process has five main component processes: prepare, collaborate, analyze, report, and decision. The process is database driven and uses the TA repository and associated databases for management and storage of TA review artifacts. This process anticipates close integration with PMO, infrastructure management, and application development organizations and processes.

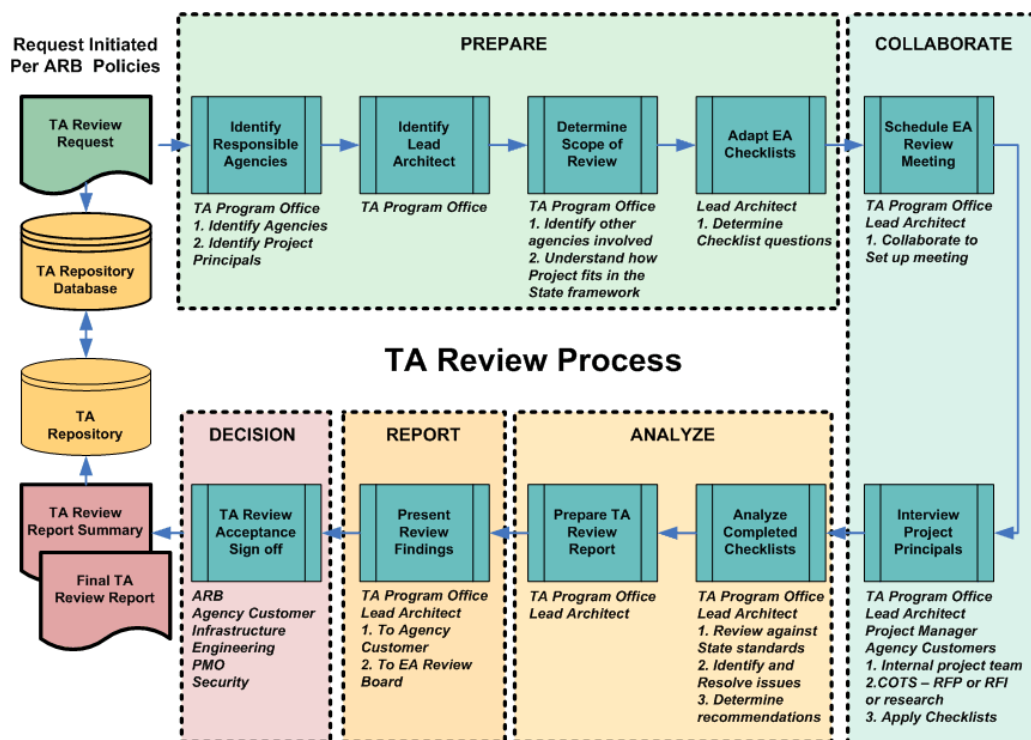


Figure 2. EA Review Process

Table 1 identifies the key EA review process roles that were identified in Figure 2.

**Table 1.** EA Review Process Roles

No.	Role	Responsibilities	Notes
1	ARB	To ensure that IT architectures are consistent and support overall business needs.	Sponsor and monitor architecture activities.
2	Project Leader (or PMO)	Responsible for project implementation.	
3	TA Program Office	To administer the architecture development and review process.	More likely to be business-oriented than technology-oriented.
4	Lead Architect	To ensure that the architecture is technically coherent and future-proof.	An IT architecture specialist.
5	Architect	One of the Lead Architect's agency technical partners.	
6	Customer	To ensure that business requirements are clearly expressed and understood.	Manages that part of the organization that will depend on the success of the IT described in the architecture.
7	Business Domain Expert	To ensure that the processes to satisfy the business requirements are justified and understood.	Knows how the business domain operates; may also be the customer.
8	Project Principals	To ensure that the architects have a sufficiently detailed understanding of customer agency processes so they can provide input to the business domain expert or to the architects.	Members of the customer's organization who have input to the business requirements that the architecture is to address.

Table 2 lists the main steps in the TA review process as detailed in Figure 1.

**Table 2.** TA Review Process Steps

No.	Action	Notes	Who
1	Request architecture review.	As directed by ARB process, policies, and procedures.	Anyone, whether IT or business-oriented, with an interest in or responsibility for the business area affected.
2	Identify responsible part of organization and relevant project principals.		TA Program Office.

## Approval Draft 1.5

3	Identify Lead Architect and other architects.		TA Program Office.
4	Determine scope of review.	Identify which other business units/departments are involved. Understand where the system fits in the State architecture framework.	TA Program Office.
5	Tailor checklists.	To address the business requirements.	TA Program Office and Lead Architect.
6	Schedule Architecture Review Meeting.		TA Program Office with collaboration of Lead Architect.
7	Interview project principals.	To get background and technical information: <ul style="list-style-type: none"> <li>For internal project: in person</li> <li>For COTS: in person or via RFP</li> </ul> Use checklists.	TA Program Office and Lead Architect and/or Project Leader, and customers.
8	Analyze completed checklists.	Review against State standards. Identify and resolve issues. Determine recommendations.	TA Program Office and Lead Architect.
9	Prepare Architecture Compliance review report.	May involve supporting staff.	TA Program Office and Lead Architect.
10	Present review findings.	To Customer To ARB	TA Program Office and Lead Architect.
11	Accept review and sign off.		TA Review Board and Customer PMO Infrastructure Engineering
12	Send assessment report and summary to TA Program Office.		Lead Architect.

## TA REVIEW CHECKLISTS

Appendix A lists sample checklist questions that can be applied to an TA review. These questions are tailored as appropriate to the specific project or architecture. The review checklists provide a wide range of typical questions that may be used in conducting TA reviews, relating to various aspects of the architecture. The organization of the questions includes the basic disciplines of system engineering, information management, security, and systems management. The checklists are based on material provided by The Open Group.

## Approval Draft 1.5

The checklists provided contain too many questions for any single review: they are intended to be tailored selectively to the project concerned (see Architecture Compliance Review Guidelines). The checklists actually used will typically be developed or selected by subject matter experts and are intended to be updated annually by interest groups in those areas.

Some of the checklists include a brief description of the architectural principle that provokes the question, and a brief description of what to look for in the answer. These extensions to the checklist are intended to allow the intelligent re-phrasing of the questions, and to give the user of the checklist a feel for why the question is being asked.

Occasionally the questions will be written, as in RFPs, or in working with a senior project architect. More typically they are expressed as part of an interview or working session with the project.

The checklists provided here are designed for use in individual architecture projects, not for business domain architecture or for architecture across multiple projects. (Doing an architecture review for a larger sphere of activity, across multiple business processes and system projects, would involve a similar process, but the checklist categories and their contents would be different.)

TA Reviews will usually be targeted for completion in a one to two week time period. Large complex reviews may take additional time.

## APPENDIX A. TA REVIEW CHECKLIST QUESTION BANK

### Hardware and Operating System Checklist

1. What is the project's lifecycle approach?
2. At what stage is the project in its lifecycle?
3. What key issues have been identified or analyzed that the project believes will drive evaluations of hardware and operating systems for networks, servers, and end-user devices?
4. What system capabilities will involve high-volume and/or high-frequency data transfers?
5. How does the system design impact or involve end-user devices?
6. What is the quantity and distribution (regional and global) of usage, data storage, and processing?
7. What applications have affinities with the project by similarities in data, application services, etc.? To what degree does the project address data affinities?
8. What hardware and operating system choices have been made before functional design of key elements of the system?
9. If hardware and operating system decisions were made outside of the project's control:
  - What awareness does the project have of the rationale for those decisions?
  - How can the project influence those decisions as system design takes shape?
10. If some non-standards have been chosen:
  - What are the essential business and technical requirements for not using existing State standards?
  - Is this supported by a business case?
  - Have the assumptions in the business case been subject to scrutiny?
11. What is your process for evaluating full lifecycle costs of hardware and operating systems?



12. How has agency financial management been engaged in evaluation of lifecycle costs?
13. Have you performed a financial analysis of the supplier?
14. Have you made commitments to any supplier?
15. Do you believe your requirements can be met by only one supplier?

#### Software Services and Middleware Checklist

1. Describe how error conditions are defined, raised, and propagated between application components.
2. Describe the general pattern of how methods are defined and arranged in various application modules.
3. Describe the general pattern for how method parameters are defined and organized in various application modules. Are [in], [in/out], [out] parameters always specified in the same order? Do Boolean values returned by modules have a consistent outcome?
4. Describe the approach that is used to minimize the number of round-trips between client and server calls, particularly for out-of-process calls, and when complex data structures are involved.
5. Describe the major data structures that are passed between major system components.
6. Describe the major communication protocols that are used between major system components.
7. Describe the marshaling techniques that are used between various system components. Describe any specialized marshaling arrangements that are used.
8. Describe to what extent the system is designed with stateful and stateless components.
9. Describe how and when state is saved for both stateful and stateless components.
10. Describe the extent to which objects are created, used, and destroyed versus reused through object pooling.

11. Describe the extent to which the system relies on threading or critical section coding.
12. Describe the approach and the internal documentation that is used internally in the system to document the methods, methods arguments, and method functionality.
13. Describe the code review process that was used to build the system.
14. Describe the unit testing that has been used to test the system components.
15. Describe the pre and post-condition testing that is included in various system modules.
16. Describe the assertion testing that is included with the system.
17. Do components support all the interface types they need to support or are certain assumptions made about what types of components will call other components either in terms of language bindings or other forms of marshaling?
18. Describe the extent to which data format problems need to be handled across different platforms.
19. Describe if numbers or strings need to be handled differently across different platforms.
20. Describe whether the software needs to check for floating-point round-off errors.
21. Describe what tools or processes have been used to test the system for memory leaks, reach ability, or general robustness.
22. Describe the layering of the systems services software. Describe the general number of links between major system components. Is the system composed of a lot of point-to-point interfaces or are major messaging backbones used instead?
23. Describe to what extent the system components are either loosely coupled or tightly coupled.
24. What requirements does the system need from the infrastructure in terms of shared libraries, support for communication protocols, load balancing, transaction processing, system monitoring, naming services, or other infrastructure services?

25. Describe how the system and system components are designed for refactoring.
26. Describe how the system or system components rely on common messaging infrastructure versus a unique point-to-point communication structure.

## Applications Checklists

### **Infrastructure (Enterprise Productivity) Applications**

1. Is there need for specialized capabilities that are not provided through the enterprise's standard infrastructure application products? For example:

- Collaboration
  - Application Sharing
  - Video Conferencing
  - Calendaring
  - E-mail
- Workflow Management
- Publishing/Word Processing Applications
  - HTML
  - SGML and XML
  - Portable Document format
  - Document Processing (Proprietary Format)
  - Desktop Publishing
- Spreadsheet Applications
- Presentation Applications
  - Business Presentations
  - Image
  - Animation
  - Video
  - Sound
  - CBT
  - Web Browsers
- Data Management Applications
  - Database Interface
  - Document Management
  - Product Data Management
  - Data Warehouses/Mart

- Program Management Applications
  - Project Management
  - Program Visibility
- 2. Describe the business requirements for enterprise infrastructure application capabilities that are not met by the standard products.

### **Application Integration Approach**

1. What integration points (business process/activity, application, data, computing environment) are targeted by this architecture?
2. What application integration techniques will be applied (common business objects [ORBs], standard data definitions [STEP, XML, etc], common user interface presentation/desktop)?

### Information Management Checklists

#### **Data Values**

1. What are the processes that standardize the management and use of the data?
2. What business process supports the entry and validation of the data? Use of the data?
3. What business actions correspond to the creation and modification of the data?
4. What business actions correspond to the deletion of the data, and is it considered part of a business record?
5. What are the data quality requirements required by the business user?
6. What processes are in place to support data referential integrity and/or normalization?

#### **Data Definition**

1. What are the data model, data definitions, structure, and hosting options of purchased applications (COTS)?
2. What are the rules for defining and maintaining the data requirements and designs for all components of the information system?

3. What shareable repository is used to capture the model content and the supporting information for data?
4. What is the physical data model definition (derived from logical data models) used to design the database?
5. What software development and data management tools have been selected?
6. What data owners have been identified to be responsible for common data definitions, eliminating unplanned redundancy, providing consistently reliable, timely, and accurate information, and protecting data from misuse and destruction?

### **Security/Protection**

1. What are the data entity and attribute access rules which protect the data from unintentional and unauthorized alterations, disclosure, and distribution?
2. What are the data protection mechanisms to protect data from unauthorized external access?
3. What are the data protection mechanisms to control access to data from external sources that temporarily have internal residence within the enterprise?

### **Hosting, Data Types, and Sharing**

1. What is the discipline for managing sole-authority data as one logical source with defined updating rules for physical data residing on different platforms?
2. What is the discipline for managing replicated data, which is derived from operational sole-authority data?
3. What tier data server has been identified for the storage of high or medium-critical operational data?
4. What tier data server has been identified for the storage of operational data?
5. What tier data server has been identified for the storage of decision support data contained in a data warehouse?
6. What Database Management Systems (DBMSs) have been implemented?

## Common Services

1. What are the standardized distributed data management services (e.g., validation, consistency checks, data edits, encryption, and transaction management) and where do they reside?

## Access Method

1. What are the data access requirements for standard file, message, and data management?
2. What are the access requirements for decision support data?
3. What are the data storage and the application logic locations?
4. What query language is being used?

## Security Checklist

1. **Security Awareness:** Have you ensured that the State security policies and guidelines to which you are designing are the latest versions? Have you read them? Are you aware of all relevant computing security compliance and risk acceptance processes? (Interviewer should list all relevant policies and guidelines.)
2. **Identification/Authentication:** Diagram the process flow of how a user is identified to the application and how the application authenticates that the user is who they claim to be. Provide supporting documentation to the diagram explaining the flow from the user interface to the application/database servers and back to the user. Are you compliant with State policies on accounts, passwords, etc?
3. **Authorization:** Provide a process flow from beginning to end showing how a user requests access to the application, indicating the associated security controls and separation of duties. This should include how the request is approved by the appropriate data owner, how the user is placed into the appropriate access-level classification profile, how the user ID, password, and access is created and provided to the user. Also include how the user is informed of their responsibilities associated with using the application, given a copy of the access agreement, how to change password, who to call for help, etc.
4. **Access Controls:** Document how the user IDs, passwords, and access profiles are added, changed, removed, and documented. The documentation should include who is responsible for these processes.

5. **Sensitive Information Protection:** Provide documentation that identifies sensitive data requiring additional protection. Identify the data owners responsible for this data and the process to be used to protect storage, transmission, printing, and distribution of this data. Include how the password file/field is protected. How will users be prevented from viewing someone else's sensitive information? Are there agreements with outside parties (partners, suppliers, contractors, etc.) concerning the safeguarding of information? If so, what are the obligations?
6. **Audit Trails and Audit Logs:** Identify and document group accounts required by the users or application support, including operating system group accounts. Identify and document individual accounts and/or roles that have super user type privileges, what these privileges are, who has access to these accounts, how access to these accounts are controlled, tracked, and logged, and how password change and distribution are handled, including operating system accounts. Also identify audit logs, who can read the audit logs, who can modify the audit logs, who can delete the audit logs, and how the audit logs are protected and stored. Is the user ID obscured in the audit trails?
7. **External Access Considerations:** Will the application be used internally only? If not, are you compliant with State external access requirements?

#### System Management Checklist

1. What is the frequency of software changes that must be distributed?
2. What tools are used for software distribution?
3. Are multiple software and/or data versions allowed in production?
4. What is the user data backup frequency and expected restore time?
5. How are user accounts created and managed?
6. What is the system license management strategy?
7. What general system administration tools are required?
8. What specific application administration tools are required?
9. What specific service administration tools are required?
10. How are service calls received and dispatched?
11. Describe how the system is uninstalled.

12. Describe the process or tools available for checking that the system is properly installed.
13. Describe tools or instrumentation that are available that monitor the health and performance of the system.
14. Describe the tools or process in place that can be used to determine where the system has been installed.
15. Describe what form of audit logs that are in place to capture system history, particularly after a mishap.
16. Describe the capabilities of the system to dispatch its own error messages to service personnel.

## System Engineering/Overall Architecture Checklists

### **General**

1. What other applications and/or systems require integration with yours?
2. Describe the integration level and strategy with each.
3. How geographically distributed is the user base?
4. What is the strategic importance of this system to other user communities inside or outside the enterprise?
5. What computing resources are needed to provide system service to users inside the enterprise? Outside the enterprise and using enterprise computing assets? Outside the enterprise and using their own assets?
6. How can users outside the native delivery environment access your applications and data?
7. What is the life expectancy of this application?
8. Describe the design that accommodates changes in the user base, stored data, and delivery system technology.
9. What is the size of the user base and their expected performance level?
10. What performance and stress test techniques do you use?
11. What is the overall organization of the software and data components?



12. What is the overall service and system configuration?
13. How are software and data configured and mapped to the service and system configuration?
14. What proprietary technology (hardware and software) is needed for this system?
15. Describe how each version of the software can be reproduced and deployed over time.
16. Describe the current user base and how that base is expected to change over the next three to five years.
17. Describe the current geographic distribution of the user base and how that base is expected to change over the next three to five years.
18. Describe how many current or future users need to use the application in a mobile capacity or who needs to work off-line.
19. Describe what the application generally does, the major components of the application, and the major data flows.
20. Describe the instrumentation included in the application that allows for the health and performance of the application to be monitored.
21. Describe the business justification for the system.
22. Describe the rationale for picking the system development language over other options in terms of initial development cost versus long-term maintenance cost.
23. Describe the systems analysis process that was used to come up with the system architecture and product selection phase of the system architecture.
24. Who besides the original customer might have a use for or benefit from using this system?
25. What percentages of the users use the system in browse mode versus update mode?
26. What is the typical length of requests that are transactional?
27. Do you need guaranteed data delivery or update, or does the system tolerate failure?

28. What are the up-time requirements of the system?
29. Describe where the system architecture adheres or does not adhere to standards.
30. Describe the project planning and analysis approach used on the project.

### **Processors/Servers/Clients**

1. Describe the client/server Applications Architecture.
2. Annotate the pictorial to illustrate where application functionality is executed.

### **Client**

1. Are functions other than presentation performed on the user device?
2. Describe the data and process help facility being provided.
3. Describe the screen-to-screen navigation technique.
4. Describe how the user navigates between this and other applications.
5. How is this and other applications launched from the user device?
6. Are there any inter-application data and process sharing capabilities? If so, describe what is being shared and by what technique or technology.
7. Describe data volumes being transferred to the client.
8. What are the additional requirements for local data storage to support the application?
9. What are the additional requirements for local software storage or memory to support the application?
10. Are there any known hardware or software conflicts or capacity limitations caused by other application requirements or situations which would affect the application users?
11. Describe how the look-and-feel of your presentation layer compares to the look-and-feel of the other existing applications.
12. Describe to what extent the client needs to support asynchronous and/or synchronous communication.

13. Describe how the presentation layer of the system is separated from other computational or data transfer layers of the system.

### **Application Server**

1. Can or do the presentation and application layers run on separate processors?
2. Can or do the application and data access layers run on separate processors?
3. Can this application be placed on an application server independent of all other applications? If not, explain the dependencies.
4. Can additional parallel application servers be easily added? If so, what is the load balancing mechanism?
5. Has the resource demand generated by the application been measured and what is the value? If so, has the capacity of the planned server been confirmed at the application and aggregate levels?

### **Data Server**

1. Are there other applications, which must share the data server? If so, please identify them and describe the data and data access requirements.
2. Has the resource demand generated by the application been measured and what is the value? If so, has the capacity of the planned server been confirmed at the application and aggregate levels?

### **COTS** (where applicable)

1. Is the vendor substantial and stable?
2. Will the enterprise receive source code upon demise of the vendor?
3. Is this software configured for the enterprise's usage?
4. Is there any peculiar data or processes that would impede the use of this software? Is this software currently available?
5. Has it been used or demonstrated for volume, availability, or service-level requirements similar to those of the enterprise? Describe the past financial and market share history of the vendor.

## System Engineering/Methods and Tools Checklist

1. Do metrics exist for the current way of doing business?
2. Has the system owner created evaluation criteria that will be used to guide the project? Describe how the evaluation criteria will be used.
3. Has research of existing architectures been done to leverage existing work? Describe the method used to discover and understand. Will the architectures be integrated? If so, explain the method that will be used.
4. Describe the methods that will be used on the project:
  - For defining business strategies.
  - For defining areas in need of improvement.
  - For defining baseline and target business processes.
  - For defining transition processes.
  - For managing the project.
  - For team communication.
  - For knowledge management, change management, and configuration management.
  - For software development.
  - For referencing standards and statements of direction.
  - For quality assurance of deliverables.
  - For design reviews and deliverable acceptance.
  - For capturing metrics.
5. Are the methods documented and distributed to each team member?
6. To what extent are team members familiar with these methods?
7. What processes are in place to ensure compliance with the methods?
8. Describe the infrastructure that is in place to support the use of the methods through the end of the project and anticipated releases.
  - How is consultation and trouble-shooting provided?
  - How is training coordinated?
  - How are changes and enhancements incorporated and cascaded?
  - How are “lessons learned” captured and communicated?
9. What tools are being used on the project? (Please specify versions and platforms). To what extent are team members familiar with these tools?
10. Describe the infrastructure that is in place to support the use of the tools through the end of the project and anticipated releases?

## Approval Draft 1.5

How is consultation and trouble-shooting provided?

How is training coordinated?

How are changes and enhancements incorporated and cascaded?

How are lessons learned captured and communicated?

11. Describe how the project will promote the re-use of its deliverables and deliverable content.
12. Will the architecture designs "live" after the project has been implemented? Describe the method that will be used to incorporate changes back into the architecture designs.
13. Were the current processes defined?
14. Were issues documented, rated, and associated to current processes? If not, how do you know you are fixing something that is broken?
15. Were existing or planned process improvement activities identified and associated to current processes? If not, how do you know this activity is not in conflict with or redundant to other Statements of Work?
16. Do you have current metrics? Do you have forecasted metrics? If not, how do you know you are improving something?
17. What processes will you put in place to gather, evaluate, and report metrics?
18. What impacts will the new design have on existing business processes, organizations, and information systems? Have they been documented and shared with the owners?

## APPENDIX B. GUIDELINES FOR CONDUCTING EA REVIEWS

- Understand clearly the objectives of those soliciting the review, and stay on track and deliver what was asked for.
- If it becomes obvious during the discussion that there are other issues that need to be addressed, which are outside the scope of the requested review, bring it up with the meeting chair afterwards. A plan for addressing the issues can then be developed in accordance with their degree of seriousness.
- Stay "scientific," rather than "We like to see large databases hosted on ABC rather than XYZ." say things like "The downtime associated with XYZ database environments is much greater than on ABC database environments. Therefore, we don't recommend hosting type M and N systems in an XYZ environment."
- Ask "open" questions; i.e., questions that do not presume a particular answer.
- There are often "hidden agendas" or controversial issues among those soliciting a review, which may not be evident up-front. A depersonalized approach to the discussions may help bridge the gaps of opinion rather than exacerbate them.
- Treat those being interviewed with respect. They may not have built the system "the way it should be," but they probably did the best they could under the circumstances they were placed in.
- Help the exercise become a learning experience for those attending and the presenters.
- Reviews should include detailed assessment activities against the architectures and should ensure that the results are stored in the ADM repository.